Comput. Math. Applic. Vol. 16, No. 1/2, pp. 93–109, 1988
Printed in Great Britain.

0097-4943/88 $3.00 + 0.00

NUMERICAL TREATMENT OF LARGE-SCALE AIR POLLUTION MODELS

Z. ZLATEV and R. BERKOWICZ
Air Pollution Laboratory, Danish Agency of Environmental Protection, Risø National Laboratory, 4000 Roskilde, Denmark

Abstract—Studying air pollution phenomena over a large space domain by a fairly general mathematical model is discussed. The space discretization of such a model leads to huge systems of ODEs. The integration algorithms used in the solution of these systems must be efficient. If a 3-D model is considered, then a vector processor is to be used. Moreover, the most time-consuming parts of the code are to be vectorized. The calculated results, concentrations of different air pollutants, must be reliable, because they have to be used by other specialists. Therefore different checks concerning the reliability of the results are carried out. In this paper it is shown that an efficient and reliable code for studying both sulphur and nitrogen pollution has been developed at the Danish Agency of Environmental Protection. Some simulation processes are also described briefly.

1. GENERAL MODEL FOR LONG-RANGE TRANSPORT OF AIR POLLUTION

The long-range transport of air pollution (LRTAP) is often described by a fairly general mathematical model consisting of a system of partial differential equations (PDEs) [1, 2]:

$$\frac{\partial c_s}{\partial t} = -\frac{\partial (uc_s)}{\partial x} - \frac{\partial (vc_s)}{\partial y} - \frac{\partial (wc_s)}{\partial z} + \frac{\partial}{\partial x}\left(K_x \frac{\partial c_s}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y \frac{\partial c_s}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_z \frac{\partial c_s}{\partial z}\right) + Q_s \quad (s = 1, 2, \ldots, q), \quad (1)$$

where

(i) $c_s(x, y, z, t)$, $s = 1, 2, \ldots, q$, is the concentration of the sth pollutant in the atmosphere,
(ii) $u(x, y, z, t)$, $v(x, y, z, t)$ and $w(x, y, z, t)$ are wind velocities along the $0x$, $0y$ and $0z$ axes,
(iii) $K_x(x, y, z, t)$, $K_y(x, y, z, t)$ and $K_z(x, y, z, t)$ are diffusitivity coefficients

and

(iv) $Q_s(x, y, z, t, c_1, c_2, \ldots, c_q)$ is a common term representing
  (a) emission sources,
  (b) different sinks
  and
  (c) chemical reactions
  concerning the sth pollutant ($s = 1, 2, \ldots, q$).

The mathematical model described by system (1) is to be considered together with some initial and boundary conditions. The boundary conditions depend on the space domain in which the model is defined and on the particular pollutants involved in the model. The boundary conditions will not be discussed in detail (a brief discussion only being given below), but it will always be assumed that some appropriate boundary conditions are attached to the system of PDEs (1).

Assume that the space domain contains the whole of Europe, as shown in Fig. 3. Assume also that either sulphur or nitrogen pollution is studied. Then the horizontal boundary conditions can be obtained by assuming that all concentrations on the boundaries are equal to zero. This is a realistic assumption for this particular case, because the regions close to the boundaries (such as, for example, the Atlantic Ocean; see the map in Fig. 3) are not polluted. Nevertheless, special care is to be taken in the presence of outgoing flows. This is done by introducing artificial sinks for outgoing flows when these come close to the boundaries. Of course, this implies that the

concentrations calculated by the model at points close to the boundaries should not be used in other studies (as they are not reliable). If only a part of the space domain given in Fig. 3 is considered, then the problem of determining a proper boundary condition becomes much more difficult (on some parts of the boundary, at least). Concentrations obtained by the present model could be used as boundary conditions in the latter case.

Assume again that either sulphur pollution or nitrogen pollution is to be studied. Assume also that system (1) is considered as a 3-D model. Then the vertical boundary conditions can be specified as follows. It is assumed that there is no flux through the upper boundary, while the boundary condition on the lower boundary is given by "*the dry deposition*" assumption: some part of the pollutants falls on the surface.

When LRTAP is studied the mathematical model is normally considered in a space domain which is a parellelepiped, defined by

$$(x, y, z) \in D = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]. \tag{2}$$

Moreover, the sides of the parallelepiped defined by the first two intervals are very long (typically, $b_1 - a_1 = b_2 - a_2 = 4800$ km), while the third side, in the vertical direction, is very short in comparison with the first two (typically, $b_3 - a_3 = 2.5$ km).

The system of PDEs (1) is integrated on a rather long time-interval

$$t \in [0, T], \tag{3}$$

where the typical length of the time-interval is 1 year.

2. SPLITTING OF THE GENERAL AIR POLLUTION MODEL

The general air pollution model unites three different physical processes:

(a) advection phenomena,
(b) diffusion phenomena

and

(c) emissions and sinks together with chemical reactions.

The advection phenomena are described by the first three terms on the r.h.s. of system (1). The transport due to the wind is defined by these terms. *The advection phenomena are the most important part of the LRTAP.*

The diffusion phenomena are described by the next three terms on the r.h.s. of system (1).

The emissions, sinks and chemical reactions are (as already mentioned) described by the terms denoted by Q_s $(s = 1, 2, \ldots, q)$.

It is natural to split system (1) according to the physical processes involved in the system. Therefore system (1) is divided into three parts (using ideas from Ref. [1]):

$$\frac{\partial c_s^*}{\partial t} = -\frac{\partial (u c_s^*)}{\partial x} - \frac{\partial (v c_s^*)}{\partial y} - \frac{\partial (w c_s^*)}{\partial z}, \quad s = 1, 2, \ldots, q; \tag{4}$$

$$\frac{\partial c_s^{**}}{\partial t} = \frac{\partial}{\partial x}\left(K_x \frac{\partial c_s^{**}}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y \frac{\partial c_s^{**}}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_z \frac{\partial c_s^{**}}{\partial z}\right), \quad s = 1, 2, \ldots, q; \tag{5}$$

and

$$\frac{\partial c_s^{***}}{\partial t} = Q_s, \quad s = 1, 2, \ldots, q. \tag{6}$$

Systems (4)–(6) are much simpler than system (1). Each of these systems has some special properties that can be exploited to design special algorithms for each of the systems. As an illustration of this fact it should be mentioned that the space discretization of system (4) very often induces operators which are skew-symmetric matrices, while the space discretization of system (5)

very often leads to operators that are positive-definite symmetric matrices. This can be exploited by choosing time-integration algorithms with large absolute stability intervals on the imaginary axis when the system of ordinary differential equations (ODEs) arising after the space discretization of system (4) is solved, while time-integration algorithms with large absolute stability intervals on the negative part of the real axis are to be selected for the time-integration of the system of ODEs arising after the space discretization of system (5).

Before discussion of the solution of systems (4)–(6) it is important to emphasize that these systems are not independent of each other. Consider a subinterval $[t_k, t_{k+1}]$ of the time-interval $[0, T]$. Assume that the values of the concentrations c_s $(s = 1, 2, \ldots, q)$ for $t = t_k$ are found in some way. Then the values of the unknown variables c_s^* in the first system (4) are calculated by using the values of c_s at $t = t_k$ as initial conditions. The values of c_s^* so found are used as initial conditions in the process of calculating the values of the unknown variables c_s^{**} in the second system (5). The third system, (6) is solved by using the solution c_s^{**} of the second system as the initial condition. Finally, the values of the unknown variables c_s^{***}, obtained as described above, are considered as approximations to the concentrations c_s at $t = t_{k+1}$. Thus, everything is prepared to perform the same kind of calculations on the next subinterval $[t_{k+1}, t_{k+2}]$ (see Ref. [1] for details).

3. SPACE DISCRETIZATION

The three systems of PDEs (4)–(6) are transformed to large systems of ODEs by performing space discretization of their r.h.s.s. The space discretization is carried out by the use of a grid

$$G = X_M \times Y_N \times Z_P, \tag{7}$$

where

$$X_M = \{x_m / x_0 = a_1, x_{m+1} > x_m \, (m = 1, 2, \ldots, M), x_M = b_1\}, \tag{8}$$

$$Y_N = \{y_n / y_0 = a_2, y_{n+1} > y_n \, (n = 1, 2, \ldots, N), y_N = b_2\} \tag{9}$$

and

$$Z_P = \{z_p / z_0 = a_3, z_{p+1} > z_p \, (p = 1, 2, \ldots, P), z_P = b_3\}. \tag{10}$$

It is not assumed that grids (8)–(10) are equidistant. However, the meteorological data are nearly always prepared on equidistant grids. These data are normally not prepared by the modellers. Moreover, the meteorological data are obtained by some kind of interpolation (the meteorological stations are not located on the grid-points of G). Therefore, as a rule, the modellers use the same grid-points as those used by the meteorologists in the process of preparing the necessary data (it is not a very good idea to interpolate interpolated data). This explains why equidistant grids are very often used in the treatment of LRTAP models. If an equidistant grid G is in use, then the distances between two adjacent grid-points are given by

$$\Delta x = (b_1 - a_1)/(M - 1) \quad \text{along the grid-lines parallel to } 0x, \tag{11}$$

$$\Delta y = (b_2 - a_2)/(N - 1) \quad \text{along the grid-lines parallel to } 0y \tag{12}$$

and

$$\Delta z = (b_3 - a_3)/(P - 1) \quad \text{along the grid-lines parallel to } 0z. \tag{13}$$

In the treatment of the model developed at the Air Pollution Laboratory of the Danish Agency of Environmental Protection the three systems of PDEs (4)–(6) are discretized by the use of (see Refs [3–6]):

$$M = N = 32, \qquad P = 9, \qquad \Delta x = \Delta y = 150 \text{ km}, \qquad \Delta z = 0.3 \text{ km}. \tag{14}$$

In some experiments $P = 17$ and $\Delta z = 0.15$ km were used.

The use of grids (8)–(10) with any discretization algorithm leads to three systems of ODEs (each of these systems contains $L = qMNP$ equations):

$$\frac{d\mathbf{g}_s^*}{dt} = \mathbf{f}_s^*(t, \mathbf{g}_s^*), \qquad s = 1, 2, \ldots, q; \tag{15}$$

$$\frac{d\mathbf{g}_s^{**}}{dt} = \mathbf{f}_s^{**}(t, \mathbf{g}_s^{**}), \qquad s = 1, 2, \ldots, q; \tag{16}$$

and

$$\frac{d\mathbf{g}_s^{***}}{dt} = \mathbf{f}_s^{***}(t, \mathbf{g}_s^{***}), \quad s = 1, 2, \ldots, q. \tag{17}$$

The quantities $\mathbf{g}_s^*$, $\mathbf{g}_s^{**}$, $\mathbf{g}_s^{***}$, $\mathbf{f}_s^*$, $\mathbf{f}_s^{**}$ and $\mathbf{f}_s^{***}$ ($s = 1, 2, \ldots, q$) are vectors with $K = MNP$ components. The r.h.s.s, $\mathbf{f}_s^*$, $\mathbf{f}_s^{**}$ and $\mathbf{f}_s^{***}$ ($s = 1, 2, \ldots, q$), are induced by the operators used in the space discretization process. The unknown functions in systems (15)–(17), $\mathbf{g}_s^*$, $\mathbf{g}_s^{**}$ and $\mathbf{g}_s^{***}$ ($s = 1, 2, \ldots, q$), are concentrations at the grid-points of G (as functions of the time variable).

The systems of ODEs (15)–(17) are very large. It is easily seen that under the assumptions made above, see equations (14), the total number of equations in systems (15)–(17) is *55,296* in the simplest case, where the number of pollutants involved in the process is $q = 2$. This indicates that one must try to exploit some specific properties of the phenomena (in order to simplify the problem) and, moreover, one should be very careful in the selection of numerical algorithms for the solution of systems (15)–(17). In the next section some assumptions are made to simplify systems (15)–(17). These assumptions are based on the specific properties of the phenomena studied. After the simplification of systems (15)–(17), the main algorithms used will be outlined.

4. SOME BASIC ASSUMPTIONS

Experience with mathematical models describing LRTAP in the atmosphere indicates that the following assumptions are realistic [7–9]:

(i) The mass conservation law, $\partial u/\partial x + \partial v/\partial y + \partial w/\partial z = 0$, holds for the lower levels of the atmosphere ($z < 2.5$ km).
(ii) The diffusivity coefficients K_x and K_y are positive constants.
(iii) The third diffusivity coefficient, K_z, is a *piecewise constant in z function*. This means that if x^*, y^* and t^* are fixed and if $0 \leqslant z \leqslant h$, then $K_z(x^*, y^*, z, t^*)$ does not vary in z. Moreover, it is also assumed that $K_z(x^*, y^*, z, t^*) = 0$ as $h < z \leqslant b_3$. The quantity $h = h(x, y, t)$ is a function that takes positive values. It is called *the mixing height* in meteorology.
(iv) There is no vertical advection. This means that $w \equiv 0$ is used in systems (1) and (4).

In the following sections it will be outlined how the above assumptions can be exploited in the selection of numerical algorithms for solving systems (15)–(17).

5. TIME-INTEGRATION ALGORITHMS

The most difficult part of the time-integration is the solution of system (15). The other two systems (16) and (17), can be treated efficiently under Assumptions (ii) and (iii) from Section 4 and when two important cases, long-range transport of sulphur and long-range transport of nitrogen, are studied.

Assumptions (ii) and (iii) from Section 4 indicate that system (16) is a system with constant coefficients along the grid-lines parallel to the $0z$ axis for $z < h$. This means that system (16) can be solved by the use of Fourier expansions of the unknown functions when Assumptions (ii) and (iii) are made. Moreover, because advection has the most important influence on the solution (the wind is the main factor in LRTAP), the number of terms in the Fourier expansions used can be

small and, thus, system (16) can be treated in an efficient way. It should be noted that there is no restriction on the time-stepsize in the solution of system (16) by the use of Fourier expansions. This means that system (16) can be treated (at each time-step) after the treatment of system (15), independently of the time-stepsize used in the solution of system (15) (see also Refs [4, 10]).

The treatment of system (17) will in general be very complicated, because this system is non-linear when chemical reactions are used. However, if sulphur or nitrogen pollution is studied, then one can assume that the chemical reactions are simple linear transformations only. This makes system (17) easily tractable and numerical algorithms, which can be used with rather large time-steps, can easily be applied. The purpose is to apply methods which can be used with the same time-stepsize as used in the solution of system (15). If this can be achieved, then system (17) can be treated (at each time-step) after the treatment of systems (15) and (16) with the same time-stepsize as used in the treatment of system (15). Such algorithms can be applied and, thus, the time-stepsize is determined by the requirements that are to be satisfied in the treatment of system (15) only.

System (15) is the most important and the most difficult part in the numerical integration process. It is important because this system represents the advection phenomena in the long-range transport and the wind is the most important factor in LRTAP. It is difficult, because the operators induced by the space discretization, in the transition from system (4) to (15), are skew-symmetric matrices. This means that methods with long intervals on the imaginary axis, h_{imag}, are needed. If predictor–corrector (PC) schemes are in use, then $h_{\text{imag}} \leqslant m$, where m is the number of formulae used in the PC scheme. It is clear that if only system (15) is to be solved, then the computational work per time-step is proportional to the number of formulae used in the PC scheme and, therefore, it does not seem to be profitable to use PC schemes with many formulae in such a case. However, after each time-step in the solution of system (15), one should also perform calculations in order to solve systems (16) and (17) with algorithms quite different from that used in the solution of system (15). Moreover, some input–output operations (or calculations needed for preparation of input–output) are to be carried out at every time-step. Finally, some known functions (e.g. the wind velocities) are to be calculated only once per step independently of the number of formulae used in the PC schemes. This analysis shows that PC schemes with several correctors can efficiently be used in the treatment of air pollution models [3, 4].

The use of PC schemes with several correctors allows the possibility of applying large time-steps in the time-integration process. However, in order to ensure an efficient application of large stepsizes, one should carry out both an accuracy check and an absolute stability check during the computations. The use of an accuracy check is well-described in connection with many codes in which it is possible to vary the stepsize. The use of an absolute stability check deserves special discussion, because the properties of the particular problem solved (the LRTAP problem) can be exploited very successfully in order to construct a very cheap and very efficient absolute stability check. The construction of an absolute stability check for system (15) will be discussed in the next section.

6. ABSOLUTE STABILITY CHECK

System (15) is only moderately stiff, but nevertheless the absolute stability requirements are clearly dominant over the accuracy requirements. Therefore the time-integration of system (15) will be carried out with many rejections when only an accuracy check is in use. Fortunately, an absolute stability check can easily be constructed and added to the algorithm. The heuristics on which such a check can be based are outlined below.

Consider first the following simple equation (as in Ref. [11]):

$$\frac{\partial c}{\partial t} = -u\frac{\partial c}{\partial x} \quad (u = \text{const}). \tag{18}$$

Define the grid

$$X_{2M+1} = \{x_m / x_0 = a_1, x_{m+1} = x_m + \Delta x \times (m = 1, 2, \ldots, 2M+1, \Delta x = (b_1 - a_1)/2M), x_{2M+1} = b_1\} \tag{19}$$

and assume that a pseudospectral (Fourier) discretization is applied to equation (18) by the use of grid (19). The result is

$$\frac{d\mathbf{g}}{dt} = -u\mathbf{S}\mathbf{g}, \qquad \mathbf{g} \in \mathbb{R}^{(2M+1)\times 1}, \qquad \mathbf{S} \in \mathbb{R}^{(2M+1)\times(2M+1)}, \tag{20}$$

where $\mathbf{S}$ is a skew-symmetric matrix induced by the space discretization operator and $\mathbf{g}$ is a vector whose components are values of the unknown function c from equation (18) at the grid-points defined by grid (19). It can be proved that if the time-step Δt satisfies

$$\Delta t \leqslant \left(\frac{h_{\text{imag}}}{|u|\pi}\frac{M}{M-1}\right)\Delta x, \tag{21}$$

then the computations carried out with the time-integration algorithm chosen and with a constant time-stepsize Δt are absolutely stable. The parameter h_{imag} is the length of the absolute stability interval on the imaginary axis for the algorithm under consideration.

It is seen from inequality (21) that if the problem is 1-D and if the wind velocity is constant, then the absolute stability leads to a bound for Δt which is proportional to the space increment Δx and depends on the inverse of the absolute value of the wind velocity. If we assume that this property holds also when the problem is multi-dimensional and when the wind velocity is variable, then similar criteria can be derived. Of course, such a criterion will not ensure absolute stability. However, experience shows that these criteria work very well in practice.

Taking into account Assumptions (i) and (iv) from Section 4 one can rewrite system (15) as follows:

$$d\mathbf{g}_s^*dt = (\mathbf{U}(t)\mathbf{S} + \mathbf{V}(t)\mathbf{P}\mathbf{S}^*\mathbf{P})\mathbf{g}_s^*, \quad s = 1, 2, \ldots, q, \tag{22}$$

where

(a) $\mathbf{U}(t)$ and $\mathbf{V}(t)$ are diagonal matrices whose diagonal elements contain values of the wind velocities at the grid-points of G at time t,

(b) $\mathbf{S}$ and $\mathbf{S}^*$ are block-diagonal matrices that are induced by the space discretization operator and whose diagonal blocks are skew-symmetric matrices

and

(c) $\mathbf{P}$ is a permutation matrix such that $\mathbf{P}^{-1} = \mathbf{P}$.

Assume that system (22) is obtained by a pseudospectral (Fourier) algorithm and that

$$u^* = \max_{t \in T_k}(\|\mathbf{U}(t)\|), \qquad v^* = \max_{t \in T_k}(\|\mathbf{V}(t)\|), \tag{23}$$

where T_k is a neighbourhood of the current time-integration point t_k.

Then one should expect the calculations to be absolutely stable if

$$\Delta t_k \leqslant \frac{h_{\text{imag}}}{\pi}\left[\frac{u^*(M-1)}{M\Delta x} + \frac{v^*(N-1)}{N\Delta y}\right]^{-1}. \tag{24}$$

Assume that $\Delta x = \Delta y$ and $M = N$. Then inequality (24) can be rewritten as

$$\Delta t_k \leqslant \frac{h_{\text{imag}}}{\pi}\frac{M}{M-1}\frac{\Delta x}{u^* + v^*} \tag{25}$$

and it is seen that the computations should be expected to be absolutely stable if the time-stepsize Δt_k used at an arbitrary step k is bounded by a term proportional to the space increment Δx and to the inverse of the sum of the norms of the wind velocity vectors in a neighbourhood of t_k. Moreover, the term of the r.h.s. of inequality (25) depends on the time-integration algorithm used and it is desirable to apply time-integration algorithms with large h_{imag} [3, 12].

The absolute stability check (25) is defined under the assumption that a variable stepsize is in use. Therefore it is necessary to verify that the use of a variable stepsize is profitable when models describing LRTAP are treated numerically. This will be done in the next section.

7. IS THE STEPSIZE VARIATION PROFITABLE?

The absolute stability check (25) indicates that when the space discretization algorithm and the time-integration algorithm are fixed (or, in other words, when Δx and h_{imag} are fixed), then the time-stepsize is optimal when it follows the variation of $u^* + v^*$ (the sum of the norms of the wind velocity vectors) in the sense that if $u^* + v^*$ is large, then the time-stepsize should be small and if $u^* + v^*$ is small, then the time-stepsize could be large. However, the use of a variable stepsize technique complicates the code. Different checks have to be made at each step in order to decide if the approximation computed is acceptable and in order to determine the best stepsize for the next step. The coefficients of the formulae used in the time-integration algorithm are dependent on the stepsize (when a variable stepsize is allowed) and have to be calculated at every time-step. This shows that the computational work per time-step is greater when a variable stepsize technique is in use. Therefore it will be profitable to apply a variable stepsize technique only if the wind velocities vary quickly in time and, moreover, if the code can be adjusted so that the variation of the stepsize follows the variation of $u^* + v^*$ in the sense described above. If $u^* + v^*$ varies quickly and if a constant stepsize is to be used, then the stepsize must be determined using the absolute stability check (25) with the largest $u^* + v^*$ for the time-interval under consideration. This shows that 2–3 stormy days may force us to carry out the whole time-integration with a very small time-stepsize. If a variable stepsize is used, then a very small time-stepsize will be used only during the stormy days and the code will quickly increase the stepsize when $u^* + v^*$ becomes smaller. This is illustrated graphically in Fig. 1. The upper curve in the figure gives the variation of $u^* + v^*$ for December 1982 (over Europe). It is seen that there is a stormy period (12–13 December). The lower curve represents the variation in the time-stepsize by the code used (code ADM developed at the Air Pollution Laboratory of the Danish Agency of Environmental Protection). All time-stepsizes are divided by 6 in order to separate the two curves. It is seen that the code is able to follow the variation of $u^* + v^*$ so that if $u^* + v^*$ is large, then the time-stepsize is small and vice versa. Moreover, it is seen that this relationship is nearly perfect. It should be noted that the code used varies not only the time-stepsize, but also the formulae during the integration process; i.e. a variable stepsize variable formula method (VSVFM) is in use. Finally, it should be stressed that if a constant

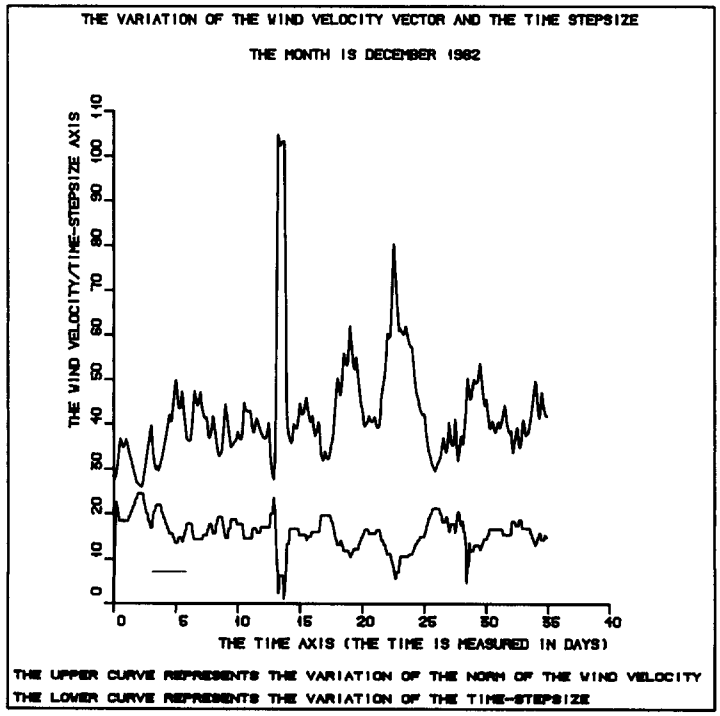

Fig. 1

stepsize is applied, then the time-stepsize *must* be very small, because of the stormy days around 12–13 December. Thus, in this case the application of a constant stepsize will be very inefficient (this is also emphasized in Ref. [5]).

8. CONSISTENCY, ZERO-STABILITY AND CONVERGENCE

The time-integration algorithm used must be consistent, zero-stable and convergent. If a constant stepsize is used, then the theory concerning these fundamental properties is well-developed. However, one must be careful when the time-stepsize and/or the formulae are allowed to vary during the integration. In general, it is not possible to guarantee the preservation of the consistency, the zero-stability and the convergence in the transition to VSVFMs. For some particular algorithms, however, this can be done. An example is described below.

Rewrite system (15) as $\mathrm{d}\mathbf{g}/\mathrm{d}t = f(t, \mathbf{g}), \mathbf{g} \in \mathbb{R}^{L \times 1}$, and let $\mathscr{F} = \{F_1, F_2, \ldots, F_m\}$ be a set, whose elements (called basic schemes) are defined by [12]

$$\mathbf{g}_k^{[0]} = \alpha_j^{[0]} \mathbf{g}_{k-1} + (1 - \alpha_j^{[0]}) \mathbf{g}_{k-1} + \Delta t \sum_{i=1}^{\nu_j^{[0]}} \beta_{ji}^{[0]} \mathbf{f}_{k-i}, \tag{26}$$

$$\mathbf{f}_k^{[0]} = f(t_k, \mathbf{g}_k^{[0]}), \tag{27}$$

$$\mathbf{g}_k^{[r]} = \alpha_j^{[r]} \mathbf{g}_{k-1} + (1 - \alpha_j^{[r]}) \mathbf{g}_{k-2} + \Delta t \beta_{j0}^{[r]} \mathbf{f}_k^{[r-1]} + \Delta t \sum_{i=1}^{\nu_j^{[r]}} \beta_{ji}^{[r]} \mathbf{f}_{k-i} \tag{28}$$

and

$$\mathbf{f}_k^{[r]} = f(t_k, \mathbf{g}_k^{[r]}), \quad r = 1, 2, \ldots, q_j. \tag{29}$$

The index j is used only to indicate that equations (26)–(29) represent the element $F_j \in \mathscr{F}$ ($j \in \{1, 2, \ldots, m\}$. At the end of the computations at step k one sets ($\mathbf{g}_k = \mathbf{g}_k^{[q_j]} \wedge \mathbf{f}_k = \mathbf{f}_k^{[q_j]}$) and continues with the calculations at step $k + 1$. It is clear that element F_j is a PC scheme with q_j correctors (not necessarily the same) [3, 12].

If a VSVFM is used, then formulae (26) and (28) should be replaced by

$$\mathbf{g}_k^{[0]} = \alpha_j^{[0]} \mathbf{g}_{k-1} + (1 - \alpha_j^{[0]}) \mathbf{g}_{k-2} + \sum_{i=1}^{\nu_j^{[0]}} \Delta t_{k-1} \beta_{ji}^{[0]}(\bar{\mathbf{h}}_{kj}) \mathbf{f}_{k-i} \tag{30}$$

and

$$\mathbf{g}_k^{[r]} = \alpha_j^{[r]} \mathbf{g}_{k-1} + (1 - \alpha_j^{[r]}) \mathbf{g}_{k-2} + \Delta t_k \beta_{j0}^{[r]}(\bar{\mathbf{h}}_{kj}) \mathbf{f}_k^{[r-1]} + \sum_{i=1}^{\nu_j^{[r]}} \Delta t_{k-1} \beta_{ji}^{[r]}(\bar{\mathbf{h}}_{kj}) \mathbf{f}_{k-i}, \tag{31}$$

where the coefficients $\beta_{ji}^{[r]}(\bar{\mathbf{h}}_{kj})$ ($k \in \{1, 2, \ldots, K\}$, $j \in \{1, 2, \ldots, m\}$, $r \in \{0, 1, \ldots, q_j\}$, $i \in \{0, 1, \ldots, \nu_j^{[r]}\}$) depend on the components of the vector

$$\bar{\mathbf{h}}_{kj} = \left(\frac{\Delta t_{k-1}}{\Delta t_k}, \frac{\Delta t_{k-2}}{\Delta t_k}, \ldots, \frac{\Delta t_{k-s_j+1}}{\Delta t_k} \right) \tag{32}$$

with

$$s_j = \max(\nu_j^{[0]}, \nu_j^{[1]}, \ldots, \nu_j^{[q_j]}), \quad j \in \{1, 2, \ldots, m\}. \tag{33}$$

Assume that:

(1) the order of each basic PC scheme $F_j \in \mathscr{F}$ is at least 1;
(2) $0 < \alpha_j^{[q_j]} < 2$, $\forall j \in \{1, 2, \ldots, m\}$;
(3) the order of the scheme defined by formulae (30) and (31) is at least equal to the order of the basic scheme $F_j \in \mathscr{F}$;

and

(4) if s_j steps are performed with the same stepsize (or, in other words, if all components of vector $\bar{\mathbf{h}}_{kj}$ are equal to one), then formulae (30) and (31) reduce to formulae (26) and (28), respectively.

Theorem 1. *If a VSVFM defined by formulae (26)–(33) is applied in the integration of a system of ODEs with* $f \in (C^{(1)}[0, T])^L$ *and if conditions (1)–(4) above are satisfied, then the VSVFM is consistent, zero-stable and convergent.* ■

The time-integration algorithm used in the integration of system (15) is based on three basic schemes of type (26)–(29) (i.e. $\mathscr{F} = \{F_1, F_2, F_3\}$) with $q_1 = 1$, $q_2 = 2$ and $q_3 = 3$. Therefore the fundamental properties (consistency, zero-stability and convergence) are guaranteed for any change in the stepsize and/or the formulae. Figure 1 shows that a very quick variation in the stepsize (needed in order to follow the variation of $u^* + v^*$) does not disturb the consistency, the zero-stability and the convergence of the VSVFM. It must be stressed that the stepsize is often changed not only very quickly, but also by a very considerable amount. This is not seen in Fig. 1 because, as mentioned before, all stepsizes are divided by 6 in order to separate the two curves.

9. ABSOLUTE STABILITY

The bound (25) shows that it is not sufficient to have an integration algorithm which remains consistent, zero-stable and convergent when the stepsize and the formulae are varied in order to follow the variation of the sum of the norms of the wind velocities $u^* + v^*$. In order to achieve an even more efficient computational process one should also try to construct time-integration algorithms with large absolute stability intervals on the imaginary axis, large h_{imag}. Algorithms with large h_{imag} will allow the use of large stepsizes. It has already been mentioned that there is a barrier, $h_{\text{imag}} \leqslant m$ (m being the number of formulae in the PC scheme used), that cannot be exceeded. It was also mentioned that the use of large h_{imag} (necessarily with $m > 1$) will *not* lead to a reduction in the computational work in the solution of system (15); the computational work per step for system (15) is proportional to the number of formulae used. However, the global integration algorithm is designed so that systems (16) and (17) [each of which contains as many equations as system (15)] can be solved using the same stepsize as that used in the solution of system (15). This means that the use of large stepsizes (which could be achieved by the use of large h_{imag} and $m > 1$) *does* lead to a decrease in the computational work in the solution of both systems (16) and (17), and, thus, to a reduction in the global computing time. There are two extra benefits when large stepsizes are used [3]:

(1) some of the known functions (e.g. the wind velocities) are to be calculated only once per time-step and independently of the number of formulae in the PC schemes used;

and

(2) at each time-step some input–output operations (or some computations related to the input–output operations; e.g. preparing some data for output in one of the following steps) have to be made.

Both benefits (1) and (2) indicate that some computations will be saved when large stepsizes are used. The scheme F_3 (see the end of Section 8), in which four formulae are used, has the largest h_{imag} ($h_{\text{imag}} = 3.26$) among the schemes used in set $\mathscr{F}$. The absolute stability region of scheme F_3 is given in Fig. 2. This scheme is the main scheme used in the integration of system (15), because it allows the use of large stepsizes. However, it should be noted that the maximal stepsize allowed in formula (25) cannot always be used; there are accuracy requirements also. If the stepsize should be reduced (because of the accuracy requirements), then it is not efficient to apply scheme F_3. Therefore the code switches to one of the other two schemes, F_2 or, if the stepsize is considerably smaller than that allowed by formula (25) with $h_{\text{imag}} = 3.26$, even F_1. The use of F_2 or F_1 instead of F_3 when the accuracy requirements are dominant leads to a reduction in the computational work in the solution of system (15). This explains why a variation in formulae is allowed. By this one saves some computational work when the stepsize has to be reduced because of the accuracy requirements. It should be noted here that scheme F_2 is more accurate than scheme F_3 and scheme F_1 is more accurate than scheme F_2. Three formulae are used in scheme F_2; the length of the absolute stability interval on the imaginary axis is $h_{\text{imag}} = 2.51$ for this scheme. The number of

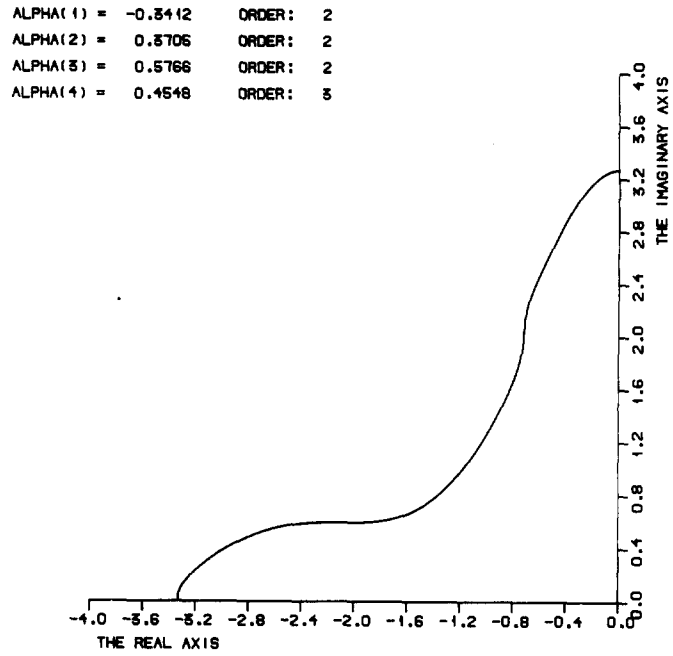

Fig. 2. The absolute stability region of the PC scheme F_3. The values of the parameters $\alpha_3^{[r]}$ ($r = 1, 2, 3, 4$) as well as the order of the four linear multistep formulae used in the PC scheme F_3 are given in the top left-hand corner.

formulae used in scheme F_1 is two; the length of its absolute stability interval on the imaginary axis is $h_{\text{imag}} = 1.62$. All three schemes were found by the use of optimization methods and $\alpha_j^{[r]}$ ($j = 1, 2, 3$; $r = 0, 1, \ldots, q_j$) as free parameters [3, 4, 12].

The use of the PC schemes F_1, F_2 and F_3 (found as described above) does not guarantee absolutely stable results in the integration of system (15). Absolute stability is guaranteed only for the special problem $\mathrm{d}\mathbf{g}/\mathrm{d}t = \mathbf{Ag}$: when $\mathbf{A}$ is a constant matrix whose eigenvalues are with non-positive real parts. However, experience shows that the schemes selected work satisfactorily. Nevertheless, some more rigorous results in this direction are highly desirable.

10. RELIABILITY OF THE ABSOLUTE STABILITY CHECK

Some extra computational work is needed when the absolute stability check (25) is carried out. At the same time this check is, as mentioned at the end of the previous section, based on heuristics. Therefore, it is necessary to test carefully the reliability of this check. When this check is in use, one should expect to obtain a computational process with large stepsizes and, what is even more important, without many rejections of time-steps (if the accuracy requirements at the end of a time-step are not satisfied, then the step is rejected, the stepsize is reduced and the calculations are repeated with the reduced stepsize). A careful study of the norms of the wind velocities during the computations showed that the code based on the absolute stability check (25) is able to carry out the integration on long time-intervals with stepsizes which are very close to the values on the r.h.s. of formula (25). This has already been illustrated in Fig. 1. The number of rejections cannot be seen from this figure. Of course, the computational process is not efficient when it is performed with many rejections of steps. However, the experiments show that the number of rejected steps is very small. As an illustration consider the following example. Model (1) was used in the study of sulphur pollution over Europe with $q = 2$. As mentioned in Section 3, the total number of equations in the splitted model is 55,296; see the discussion after systems (15)–(17). The length of the time-integration interval was 372 days; *from 25 December 1978 to 1 January 1980.* Some

characteristics concerning the computational aspects of this run are given below:

Number of equations in system (15)	18,432
Time-integration interval (in hours)	8928
Number of successful time-steps	5338
Average time-stepsize (in hours)	1.67
Number of rejected time-steps	21
Computing time (in seconds on CRAY X-MP)	1827

It is seen that only 0.39% of the successful steps are rejected. This fact illustrates the great efficiency of the absolute stability check in the solution of system (15). It is possible to handle numerically very large models only because the solution of systems (16) and (17) does not cause any stability problems (when the stepsize used in the integration of system (15) is also applied in the integration of these two systems) and because the stability of the computations during the solution of system (15) is controlled in a very reliable way.

11. VECTORIZATION OF CODE ADM

Systems (15)–(17) are very large. It was mentioned several times that the total number of equations in these systems is 55,296 even in the simple case where sulphur or nitrogen pollution is studied and the number of pollutants involved is only two. These systems could be treated numerically over a long time-interval only on vector processors. Moreover, it is necessary to vectorize the code in order to achieve the so-called super-vector level of performance, 50–160 MFLOPS [million floating point operations (multiplications or additions) per second], in the most time-consuming parts of the model. The code used in the numerical treatment of the model, code ADM, is vectorized. The most time-consuming parts, the discretization of the space derivatives, is carried out at a very high level of performance. Of course, there are parts of the model that cannot be vectorized, or can be vectorized but a super-vector level of performance cannot be achieved. This is certainly true for the input–output operations (it should be mentioned, for example, that many data are read at the end of every period of 6 h), for the loops where basic functions are called (e.g. ABS, EXP, AMAX1 etc.). Nevertheless, the results given in the previous section show that the code performs very efficiently. Comparisons with runs on an IBM 3081 (over a time-interval of only 1 month) indicate that on a CRAY X-MP the code performs about 25 times quicker. It should be stressed here that the overall performance for the complete run (because precisely this is of interest in this situation) is compared. We do not compare the performance on parts of the run (e.g. solving systems of linear algebraic equations, matrix products, matrix–vector multiplications, fast Fourier transforms etc.), but it should be emphasized that such operations are performed very efficiently. The most important facts in this context can be summarized as follows. *It is not possible to carry out runs over long time-intervals (at least equal to 1 year) with the 3-D model on the sequential computer at our disposal (an IBM 3081). Only runs on a short period (1 month) or runs with the 2-D model have been performed on the IBM 3081 computer. There is no problem when the vectorized version of code ADM is used on a CRAY X-MP.*

12. RELIABILITY OF THE CONCENTRATIONS CALCULATED BY THE MODEL

The concentrations calculated by the model are to be used in different studies by other specialists. Therefore it is absolutely essential to check the reliability of these concentrations. This can be done by making comparisons between calculated and measured concentrations. A set of measurement stations located in different European countries has been established within an international project, EMEP (European Monitoring and Evaluation Program), in which practically all European countries participate. These stations are shown in Fig. 3. Many comparisons of calculated and measured concentrations were performed at the Air Pollution Laboratory of the Danish Agency of Environmental Protection. A few of the results obtained from these comparisons are given in Figs 4–8.

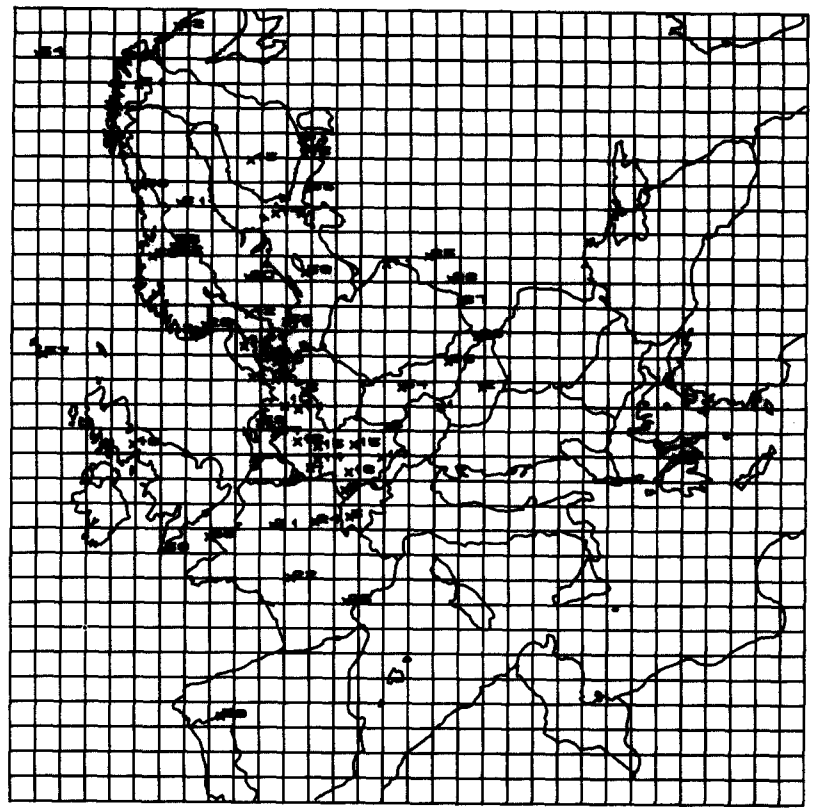

Fig. 3. Map of Europe with the grid-net used. The EMEP stations working in 1979 are given on the plot. Polar stereographic projection through the 60th degree is applied. A rotation of 32° about the North Pole is also performed.

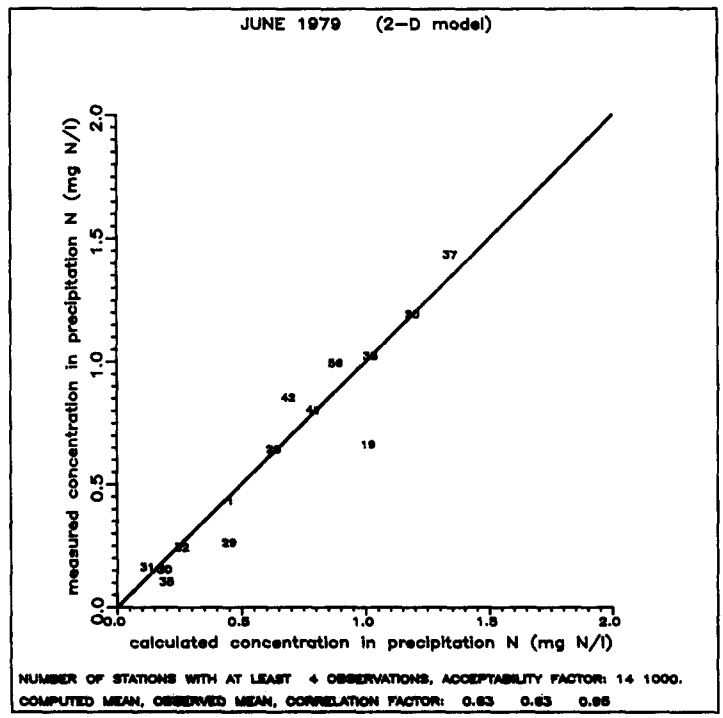

Fig. 4

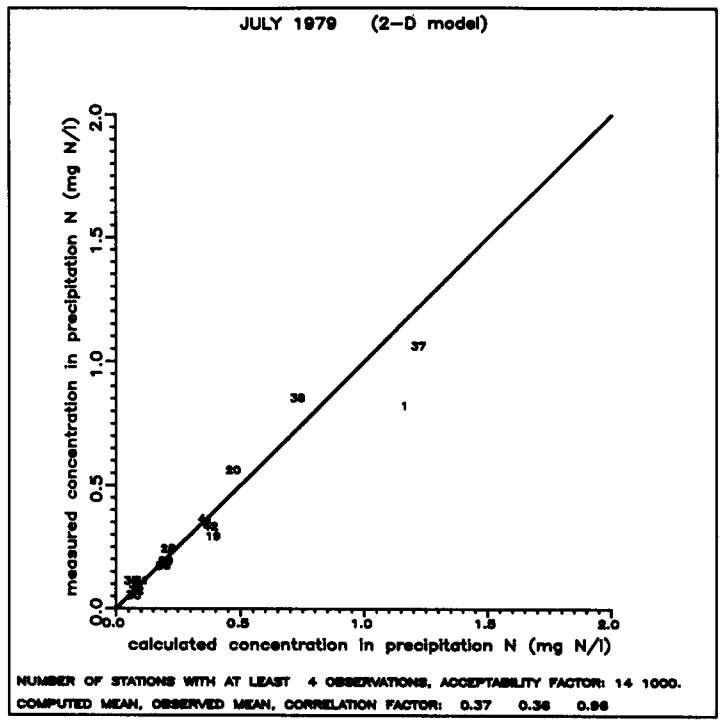

Fig. 5

In Figs 4–6 each measurement station is denoted by its identification number from Fig. 3. The abscissa of each station is the calculated concentration, while the measured concentration is the ordinate. Figures 7 and 8 present the variations in the calculated and measured concentrations for one of the EMEP stations. It should be mentioned that the 2-D model has been used in most of

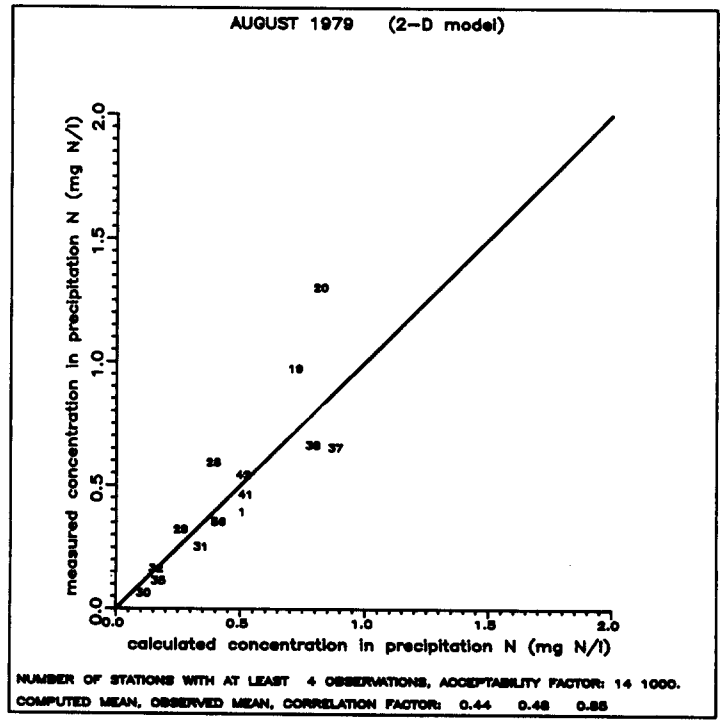

Fig. 6

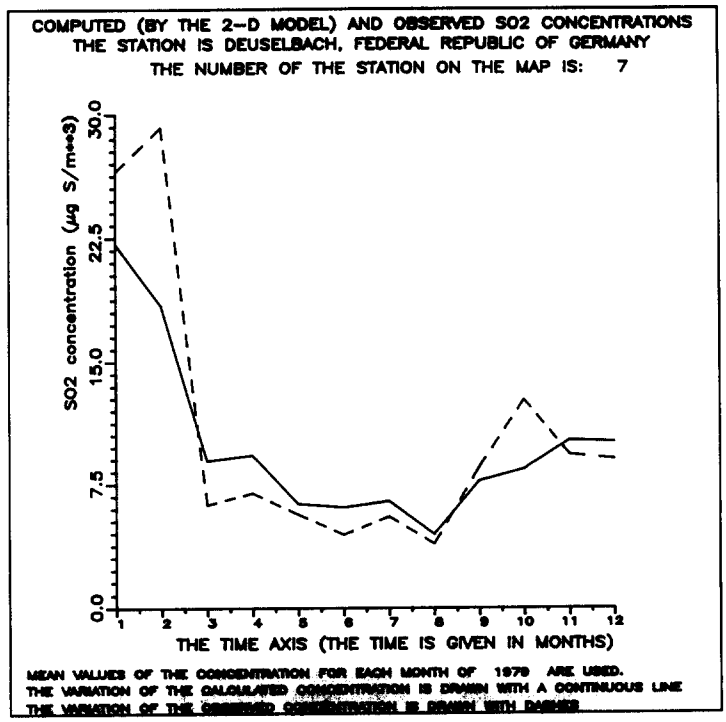

Fig. 7

the experiments. *The results indicate that the model is able to calculate reliable* concentrations. Indeed, the discrepancies between the calculated and measured concentrations are comparable (of the same order of magnitude) with the uncertainties in the measurements and, according to the entropy law, one should not expect anything better. More results can be found in Refs [4–9].

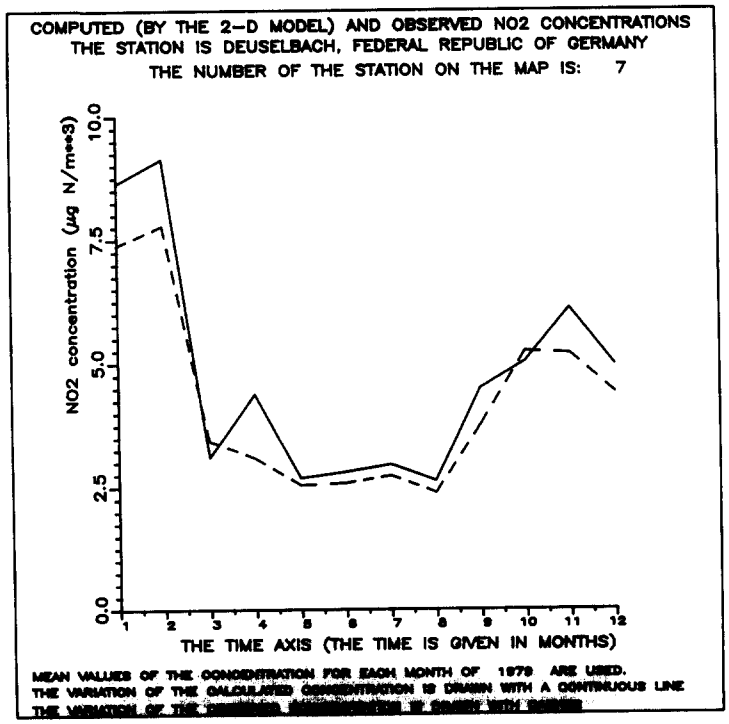

Fig. 8

13. SIMULATION OF DIFFERENT SITUATIONS

The fact that the model provides reliable concentrations can be exploited to organize experimental and simulation processes that cannot be performed in another way. As an illustration, an experiment, in which the model has been applied to calculate the depositions of nitrogen and sulphur in Denmark and the surrounding countries caused by the Danish emission sources alone, is described below. Two runs of the code were performed. In the first run all European emission sources were activated, while only the Danish emission sources were used in the second run. It is clear that if the model gives reliable results, then the amount of deposition in Denmark and the surrounding countries that is caused by the Danish emission sources alone can be evaluated. Some results concerning the depositions of sulphur and nitrogen are given in Figs 9 and 10. More numerical results are given in Ref. [8]. It should be emphasized here that a 2-D version of the model was used to obtain the results given in the previous two sections.

14. BIBLIOGRAPHICAL NOTES

Model (1) or similar models are discussed, for example, in Refs [1, 2, 4]. The Lagrangian idea, leading to trajectory methods, is also commonly used in air pollution environments [13].

Splitting procedures similar to that introduced in Section 2 are studied in Ref. [1].

The particular algorithms used in the space discretization of the model developed at the Air Pollution Laboratory of the Danish Agency of Environmental Protection are studied in detail in Refs [4–6, 10, 11, 14, 15]. The particular time-integration algorithms applied are also studied in detail in Refs [3, 4, 12, 16–18].

The time-integration algorithms are used as VSVFMs (because this mode is the most efficient way to treat air pollution transport models; especially LRTAP models). The fundamental properties of the particular VSVFMs used are studied in Refs [3, 4, 12, 16, 17]. Other studies of such methods are presented in Refs [19–22].

The concept of absolute stability used in this paper is defined in many books on ODEs [e.g. 23].

The absolute stability check used is very simple and very cheap, but it can be applied only in connection with the particular models treated here. There exist more general checks, but they are

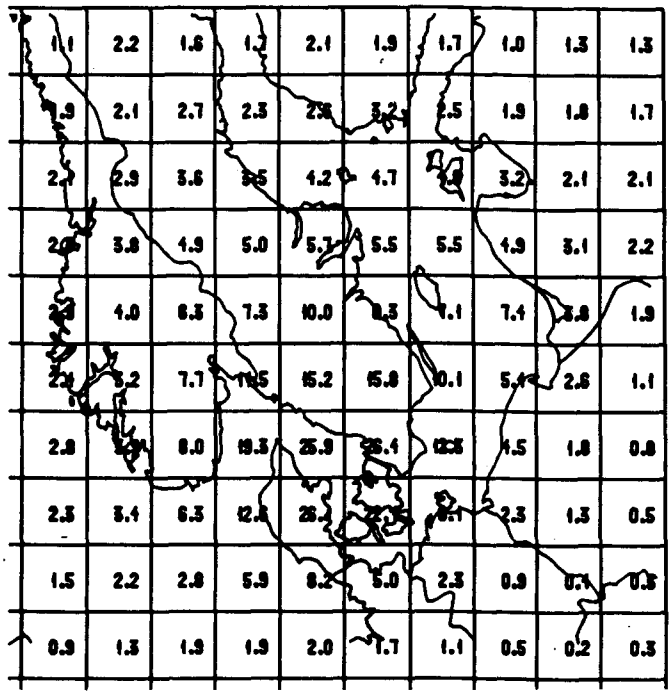

Fig. 9. Total sulphur deposition for 1979: the contributions of the Danish sources are given as percentages.

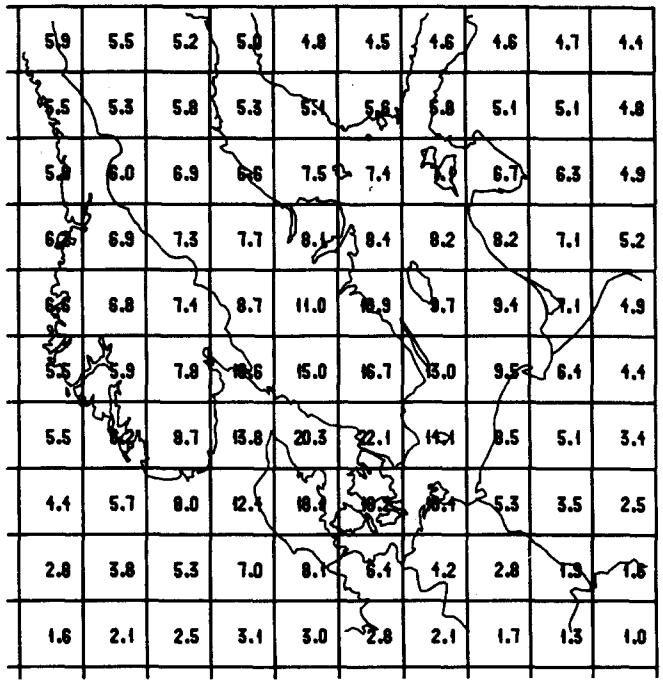

Fig. 10. Total nitrogen deposition for 1979: the contributions of the Danish sources are given as percentages.

based on an evaluation of the largest eigenvalue of the Jacobian matrix of the system of ODEs, and therefore will be much more expensive for the particular system (15). Some more general absolute stability checks are considered in, for example, Refs [24, 25].

Many general-purpose codes based on VSVFMs have been developed during the last 15 years [e.g. 26–33]. A general-purpose code is not suitable for the problems studied in this paper; this is also explained in Refs [3, 5]. Indeed, implicit time-integration methods are prohibitive both because the problem is only moderately stiff and because the systems solved are very large. Explicit methods, including here the PC schemes, are effective only if the absolute stability intervals on the imaginary axis are sufficiently large. However, the only explicit time-integration algorithms used in the general-purpose codes are based on Adams formulae, which have rather poor absolute stability properties on the imaginary axis. This explains why special PC schemes were constructed for the air pollution problems studied.

Some experimental results *are* given in this paper. Many other results, presented in a systematic way, can be found elsewhere [7–9].

Acknowledgements—Following the constructive comments of the referee the authors have been able to improve the presentation of the results considerably. Therefore the authors would like to thank the referee for his/her careful work on the manuscript and for his/her helpful suggestions for improvements.

REFERENCES

1. G. I. Marchuk, *Mathematical Modeling in the Problem of the Environment*. Nauka, Moscow (1982) (in Russian). English translation, *Studies in Mathematics and Applications*, No. 16. North-Holland, Amsterdam (1982).
2. G. J. McRae, W. R. Goodin and J. H. Seinfeld, Numerical solution of the atmospheric diffusion equation for chemically reacting flows. *J. comput. Phys.* **45**, 1–42 (1982).
3. Z. Zlatev, Application of predictor–corrector schemes with several correctors in solving air pollution problems. *BIT* **24**, 700–715 (1984).
4. Z. Zlatev, Mathematical model for studying the sulphur pollution over Europe. *J. comput. appl. Math.* **12**, 651–666 (1985).
5. Z. Zlatev, R. Berkowicz and L. P. Prahm, Implementation of a variable stepsize variable formula method in the time-integration part of a code for treatment of long-range transport of air pollutants. *J. comput. Phys.* **55**, 278–301 (1984).

6. Z. Zlatev, R. Berkowicz and L. P. Prahm, Package ADM for studying long-range transport of pollutants in the atmosphere. In *PDE Software: Modules, Interfaces and Systems* (Edited by B. Engquist and T. Smedsaas), pp. 153–169. North-Holland, Amsterdam (1984).
7. Z. Zlatev and R. Berkowicz, Experiments with a mathematical model for nitrogen pollutants. Technical Note, Air Pollution Lab. Danish Agency of Environmental Protection, Risø National Lab., Roskilde (1986).
8. Z. Zlatev, R. Berkowicz and L. P. Prahm, Studying the sulphur pollution over Europe. Report No. MST LUFT-A98, Air Pollution Lab., Danish Agency of Environmental Protection, Risø National Lab., Roskilde (1985).
9. Z. Zlatev, R. Berkowicz and L. P. Prahm, Studying the nitrogen pollution over Europe: I. Description of the model and some simple tests. Report No. MST LUFT-A99, Air Pollution Lab., National Agency of Environmental Protection, Risø National Lab., Roskilde (1985).
10. Z. Zlatev, R. Berkowicz and L. P. Prahm, Three-dimensional advection-diffusion modelling for regional scale. *Atmos. Envir.* **17,** 491–499 (1983).
11. Z. Zlatev, R. Berkowicz and L. P. Prahm, Testing subroutines solving advection–diffusion equations in atmospheric environments. *Comput. Fluids* **11,** 13–38 (1983).
12. Z. Zlatev, Variable stepsize variable formula methods based on predictor–corrector schemes. *Appl. numer. Math.* **1,** 395–416 (1985).
13. A. Eliassen, The OECD study of long range transport of air pollutants: long range transport modelling. *Atmos. Envir.* **12,** 479–487 (1978).
14. Z. Zlatev, R. Berkowicz and L. P. Prahm, Choice of time-integration scheme in pseudospectral algorithm for advection equations. In *Computational Methods for Fluid Dynamics* (Edited by K. W. Morton and M. J. Baines), pp. 303–321. Academic Press, London (1982).
15. Z. Zlatev, R. Berkowicz and L. P. Prahm, Stability restrictions on time-stepsize for numerical integration of first-order partial differential equations. *J. comput. Phys.* **51,** 1–27 (1983).
16. Z. Zlatev, Stability properties of variable stepsize variable formula methods. *Num. Math.* **31,** 175–182 (1978).
17. Z. Zlatev, Zero-stability properties of variable stepsize variable formula methods. *Num. Math.* **37,** 157–166 (1981).
18. Z. Zlatev, Consistency and convergence of general linear multistep variable stepsize variable formula methods. *Computing* **31,** 47–67 (1983).
19. M. Crouzeix and F. J. Lisbona, The convergence of variable-stepsize, variable-formula methods. *SIAM Jl numer. Analysis* **21,** 512–534 (1984).
20. C. W. Gear and K. W. Tu, The effect of variable mesh on the stability of multistep methods. *SIAM Jl numer. Analysis* **11,** 1024–1043 (1974).
21. C. W. Gear and D. S. Watanabe, Stability and convergence of variable multistep methods. *SIAM Jl numer. Analysis* **14,** 1044–1053 (1974).
22. R. D. Grigorieff, Stability of multistep-methods on variable grids. *Num. Math.* **42,** 359–377 (1983).
23. J. D. Lambert, *Computational Methods in Ordinary Differential Equations.* Wiley, London (1973).
24. L. R. Petzold, Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. Report SAND80-8230, Sandia National Labs, Livermore, Calif. (1980).
25. L. F. Shampine, Type-insensitive ODE codes based on implicit A-stable formulas. *Maths Comput.* **36,** 499–510 (1981).
26. K. Burrage, J. C. Butcher and F. H. Chipman, An implementation of singly implicit Runge–Kutta methods. *BIT* **20,** 326–340 (1980).
27. A. R. Curtis, The FACSIMILE numerical integrator for stiff initial value problems. In *Computational Techniques for Ordinary Differential Equations* (Edited by I. Gladwell and D. K. Sayers), pp. 42–87. OUP, London (1980).
28. C. W. Gear, The automatic integration of ordinary differential equations. *Communs Ass. comput. Mach.* **1,** 176–179 (1971).
29. C. W. Gear, Algorithm 407, DIFSUB for solution of ordinary differential equations. *Communs Ass. comput. Mach.* **14,** 185–190 (1971).
30. A. C. Hindmarsh, ODEPACK, a systemized collection of ODE solvers. In *IMACS Transactions on Scientific Computation,* Vol. 1 (Edited by R. S. Stepleman). North-Holland, Amsterdam (1983).
31. F. T. Krogh, VODQ/SVDQ/DVDQ—variable order integrators for the numerical solution of ordinary differential equations. Report, Jet Propulsion Lab., Pasadena, Calif. (1969).
32. L. F. Shampine and M. K. Gordon, *Computer Solution of Ordinary Differential Equations.* Freeman, San Francisco, Calif. (1975).
33. Z. Zlatev and P. G. Thomsen, Automatic solution of differential equations based on linear multistep methods. *ACM Trans. math. Softw.* **5,** 401–414 (1979).